A New Solution and Concrete Implementation Steps for Realizing Truly Universal Artificial Intelligence

Authors: Chen Yongcong, Ting Zeng, Zhang Jun, Yongcong Chen, Ting, Zeng

Date: 2023-08-15T00:00:00+00:00

Abstract

At present, the mainstream artificial intelligence generally adopts the technical path of "attention mechanism + deep learning" + "reinforcement learning". It has made great progress in the field of AIGC (Artificial Intelligence Generated Content), setting off the technical wave of big models[2][13]. But in areas that need to interact with the actual environment, such as elderly care, home nanny, agricultural production, and vehicle driving, trial and error are expensive and a reinforcement learning process that requires much trial and error is difficult to achieve. Therefore, in order to achieve Artificial General Intelligence(AGI) that can be applied to any field, we need to use both existing technologies and solve the defects of existing technologies, so as to further develop the technological wave of artificial intelligence. In this paper, we analyze the limitations of the technical route of large models, and by addressing these limitations, we propose solutions, thus solving the inherent defects of large models. In this paper, we will reveal how to achieve true AGI step by step.

Full Text

Preamble

A New Solution and Concrete Implementation Steps for Realizing Truly Universal Artificial Intelligence

Chen Yongcong
New Millennium Future Technology Co., Ltd., Beijing
yongcongchen@sina.com

Zeng Ting
Tsinghua University, Beijing
ceng-t@mail.tsinghua.edu.cn

Zhang Jun
Shenyang Normal University, Shenyang

Abstract

Currently, mainstream artificial intelligence adopts the technical path of "attention mechanism + deep learning + reinforcement learning." This approach has achieved remarkable progress in the field of AIGC (Artificial Intelligence Generated Content), sparking the wave of large-scale models [**?**][**?**]. However, in domains requiring interaction with the physical environment—such as elderly care, domestic assistance, agricultural production, and autonomous driving—trial-and-error is prohibitively expensive, making reinforcement learning, which requires extensive exploration, impractical. To achieve Artificial General Intelligence (AGI) applicable to any field, we must leverage existing technologies while addressing their fundamental limitations to advance the next wave of AI innovation. This paper analyzes the constraints of current large model architectures and proposes solutions that rectify these inherent defects, revealing a step-by-step path to true AGI.

Keywords: artificial general intelligence, AGI, AIGC, reinforcement learning, large model, ChatGPT

Introduction

The current mainstream AI model has ignited the spark of artificial general intelligence [**?**], yet it remains far from truly universal AI. Where exactly is the ceiling of present-day AI models? Can the combination of "attention mechanism + deep learning + reinforcement learning" genuinely achieve AGI? We contend that current large AI models suffer from three critical flaws that prevent them from reaching this goal.

First, they cannot solve problems independently. For instance, AI does not proactively offer assistance when it observes its owner fall [**?**]. This stems from machines lacking intrinsic needs, making it impossible for them to generate their own goals. Without self-derived goals, machines cannot spontaneously create tasks. In essence, large models do not create novel processes—they remain fundamentally programming platforms whose language is natural language [**?**]. No matter how many high-level functions we integrate or tools we embed, large models cannot spontaneously generate new processes. All their processes are preset, either through explicit programming or statistical patterns from data. Both approaches essentially apply "out-of-set processes to all problems" [**?**][**?**][**?**][**?**][**?**]. Regardless of how many conditional branches exist in these processes, they are predetermined and pre-existing, not created by the machine for specific tasks. Consequently, decision-making machines operating on predetermined processes exhibit "bookworm" intelligence—rigid and ill-equipped to handle the endless unexpected situations of real life. This represents the current dilemma of artificial intelligence.

Second, knowledge cannot be updated in real time. Contemporary AI relies on massive data training, preventing real-time knowledge updates. For machines interacting with their environment, real-time knowledge acquisition is crucial because environmental interaction is precisely how machines gain new knowledge. Without real-time updates, machines facing identical inputs will repeatedly make the same mistakes [**?**].

Third, they cannot be applied to domains requiring real-world interaction. In areas like autonomous driving, household chores, and patient care, machines must build knowledge about interactive decision-making between their actions and the external environment. These domains do not permit extensive trial-and-error, making it impossible for machines to use reinforcement learning to construct decision-making knowledge through real-world interaction [**?**][**?**]. We envision a future where every family has a robot nanny, all vehicles drive autonomously, and robots handle all industrial, agricultural, and service work while humans enjoy life's beauty. Current AI technology cannot realize this vision.

2.1 How to Describe Information Contained in a Matrix?

Although a matrix may contain substantial information, we can express all of it by establishing a complete set of coordinate basis clusters. If this basis set is complete and orthogonal, it can represent all information in the matrix. If the coordinate matrix clusters are complete but not orthogonal, they can still express any information, though less efficiently. If the basis clusters are incomplete, some vectors cannot be expressed, necessitating dimension expansion.

When basis coordinate clusters are eigen-orthogonal, we achieve full vector information representation with the most concise coefficients. For non-orthogonal but complete bases, the resulting coefficient matrix becomes sparse (highly expressive). If we only care about common information patterns in the matrix, we can use these patterns as coordinate bases. While inefficient for overall information representation, this approach efficiently expresses common information (yielding a sparse coefficient matrix). Therefore, when living in an information matrix space and needing to identify, analyze, and generate diverse information, the most crucial task is finding a set of basis coordinates in that information matrix space.

2.2 How Do Humans Create Knowledge?

The information humans can recognize constitutes only a tiny fraction of world information due to our limited resolution. The spatiotemporal relationship between atom arrangements on a blade of grass is information, yet we do not perceive it. Through evolution, humans developed token recognition ability. Tokens are the smallest information units commonly used by humans—a straight line, for instance. Tokens themselves constitute a "world model," the fundamental building blocks for constructing humanity's magnificent knowledge palace.

Evolution endowed us with "pattern recognition" capability that adopts models like Tokens to identify surrounding information, dramatically improving information recognition energy efficiency. This is evolution's gift.

We therefore treat minimal information units—points, lines, surfaces, colors, textures, curvature, syllables, tones, symbols, touch, temperature, direction, etc.—as Tokens. Humans live in a four-dimensional Token matrix (three spatial dimensions + time). For us, this 4D Token matrix contains all knowledge from the Big Bang to the present.

Humans gradually assign symbols (linguistic symbols) to represent common Token combinations, forming concepts. We use concepts to describe any information in the matrix through conversation and writing. These concepts serve as coordinate basis clusters in our information space matrix. Under such basis clusters, coefficient expressions for common information (vectors) become sparse. For example, "investor" represents "human family, wealthy, wants to make more money, seeks help to earn, takes risks, signs agreements, shares income…"

Concepts contain common Token combinations and linguistic symbols. Since linguistic symbols appear more frequently and are more representative, they often become the primary entry point for concepts. Human concepts are clearly non-orthogonal—we habitually treat frequent Token combinations as concepts. Token combinations within concepts may be non-overlapping, partially overlapping, or fully inclusive. Common Tokens existing across numerous entities are highly representative but low-resolution, representing abstract concepts. Adding more Tokens to abstract concepts creates more concrete concepts with reduced scope and higher resolution.

While such coordinate basis clusters inefficiently express all information, they efficiently express common information. Concepts like "cat" and "dog" may share numerous non-orthogonal common Tokens, yet this representation proves highly efficient for daily human information processing. Crucially, this enables information generalization: properties of things are essentially combinations of constituent Token properties. "Cat" represents a common spatiotemporal arrangement of Tokens spanning language, text, sound, images, actions, touch, and other multimodal matrix elements. Some matrix elements carry higher weight due to greater commonality, potentially belonging to the "animal" concept. "Animal" contains fewer elements, is more broadly applicable, and allows direct reuse of shared Tokens between "cat" and "dog." This is the process of information generalization and the origin of intelligence.

2.3 How Does Deep Learning Work?

In deep learning, each neural network layer's coefficients imply a set of coordinate bases. The transformation from layer A to layer A+1 is essentially a basis transformation process from layer A's coefficient matrix (expressing information with layer A's underlying coordinate basis cluster) to layer B's coefficient matrix

(expressing information with layer B's underlying coordinate basis cluster), followed by partial information compression or discarding via nonlinear functions. Deep learning's essence is using trial-and-error to find suitable coordinate basis clusters that "dilute" useful information coefficients in the input.

Residual networks reduce information loss per layer, enabling multiple transformations and increasing the probability of finding optimal basis clusters. Regularization forces intermediate layer hidden bases toward orthogonality to avoid dimensional interference and local optima, achieved by forcing the middle layer's coefficient matrix toward a sparse matrix.

The high-dimensional features created by deep learning form coordinate basis clusters in the information matrix. However, they lack the constraint of "using common Token combinations." Through trial-and-error under error constraints, the established coordinate basis clusters favor efficient orthogonal representation—effective for overall useful information expression but divergent from human habits (humans only need efficient expression of common information). Consequently, deep learning and human communication remain "chicken and duck talk," fundamentally mismatched.

In large models, the attention mechanism predicts the commonality (occurrence probability) of locally statistical knowledge obtained through pre-training, becoming the preferred coordinate basis cluster for identifying, analyzing, and generating various Token combinations. Such basis clusters better align with human habits, enabling language-based communication between large models and humans. While overall expression efficiency may not be maximal, it proves more efficient for common information expression.

2.4 What Is the Nature of the Attention Mechanism?

The attention mechanism's core is Bayesian inference (conditional probability), conceptually "the probability of N Token combinations AND the probability of M Token combinations." In human language, N Token combinations are nearly endless, and M Tokens are infinite, making it impossible for machines to solve "given N Token combinations, find M Token combinations' probability." In multimodal contexts, this problem becomes even more pronounced. Machines can only speculate on M Token combinations' probability after N Token combinations based on limited statistics—this is the attention mechanism's nature. The weight matrix obtained through pre-training represents finite statistical knowledge. The attention mechanism uses this limited statistical knowledge to introduce current Tokens with M Token combinations' probability. If some Tokens in N+M Tokens have high weight, they frequently co-occur and are more likely to be common Token combinations.

This is the attention mechanism's central mechanism: finding common Token arrangements. Its essence can be viewed as Bayesian inference combined with neural networks. In language models, "common Token combinations" are "com-

mon words," containing organizational structures similar to grammar and specific "common phrases"—far exceeding human-scale common phrases.

The attention mechanism closely resembles human learning. When we learn information from a book, "read thin first, read thick later" follows the same principle. "Reading thin first" summarizes framework information—a compression process—while "reading thick again" adds different details (combining with other vectors to form new knowledge) on this framework—an information generation process [**?**][**?**][**?**].

2.5 How Do Large Models Work?

In large models, when information is input, the attention mechanism's inference process projects the input vector onto the coordinate basis cluster. The attention mechanism's weights are coordinate values [**?**]. The input Tokens project vectors into the weight matrix in the first layer—a vector decomposition process. The second layer projection combines and weights input Tokens after the pre-trained weight matrix Tokens combination, forming a projection process (combination-to-combination projection). Multi-layer attention mechanisms create multiple projection decomposition processes from input Token combinations to pre-trained Token combinations.

The final attention layer's weight coefficient matrix and its implied coordinate basis cluster (using common Token combinations as the basis cluster) together form the re-description of input information (self-attention mechanism). Therefore, large models work as follows: (1) they use pre-trained weight matrix Token combinations as the basis cluster, where the weight matrix represents local statistical knowledge obtained from training data through trial-and-error; (2) they employ the attention mechanism to project input Token combinations onto weight Token combinations (vector decomposition), with inference process weights as coordinate values; (3) with vector components, they find numerous adjacent vectors, and the next vectors corresponding to these adjacent vectors become output vectors. Vector proximity relationships manifest as output vector probabilities.

Thus, large models are autoregressive prediction models that perform coordinate basis cluster transformation on the original input basis (where each Token is a dimension). They convert the original basis cluster ("every Token is a dimension") into a coordinate basis cluster ("common Token combinations as dimensions"), then perform autoregressive prediction.

2.6 Why Do Big Models Emerge? When Does Emergence Occur?

Why do big models exhibit "emergence"? Consider an American arriving in China who can complete accurate translation through extensive common background information (personal needs, social structure, etc.) and moderate

Chinese-English comparisons. Large models resemble aliens lacking common background information with humans, seeing only how human information connects. They must extract human information connection patterns to predict information development. Initially, with insufficient samples, their "information framework" diverges significantly from the human framework, causing continuous mistakes and groping in the dark. As sample sizes increase, alignment probability between their framework and human framework rises. However, this is non-linear: before reaching a threshold, progress in deciphering ancient language remains minimal; at a certain node, when accuracy reaches the threshold, the entire decryption process accelerates dramatically and completes rapidly.

This is the "emergence" phenomenon. Machines do not "emerge" intelligence but rather discover the correct "common way to combine Tokens." Since evaluation criteria follow human standards, capability emerges when the machine's basis approaches the human basis.

2.7 Can RLHF Ultimately Solve Large Model Problems?

Large models face two serious problems:

2.7.1 The Hallucination Problem [?]

Large models' core capability is transforming input information to a coordinate basis cluster composed of common Token combinations (vector projection decomposition)—an information space basis transformation process. They then use the obtained coefficient matrix (attention mechanism inference weights) to find multiple similar "pre-trained vectors" (component-weighted comparison). Following the mapping relationship obtained from pre-training, they find the "next vector" and select one for output. This is the autoregressive prediction process, the working mechanism of GPT-class large models. Large models optimize "parameters," where each parameter corresponds to a Token combination set. While appearing to optimize network parameters, they essentially optimize common Token combinations—seeking optimal coordinate basis clusters. Each neural network layer's coefficients correspond to underlying basis coordinate clusters.

Large models possess only "common Token combinations" derived from massive data, lacking factual memory. Facing input Tokens, they can only decompose input information onto the "coordinate basis cluster" and obtain next Tokens with different probabilities. This iterative process is inherently creative. If facts are "common," they are retained as "common Token combinations." If facts are not retained this way or lack sufficient weight, the machine creates information. Since GPT itself generates information, the hallucination problem is inherent to its function [**?**][**?**][**?**], leaving GPT with no solution.

For example, machines observe that many journalist profiles contain links to other articles or past awards. If machines detect this information organization

pattern, it becomes a mapping from "framework" to "framework." When input information contains a similar framework but only the reporter's name differs, the machine can map "framework + details" to "framework + details," generating numerous web links or awards in output. However, these links and awards are constructed by mapping "framework + other vectors" to "framework + other vectors" and probably do not exist at all!

Many attempt to solve large model hallucination by plugging in "vector databases" for factual knowledge queries. This represents another version of implementing general AI through encyclopedias. Whether "vector database" or "knowledge graph," this cannot solve hallucination because such knowledge is plug-in and cannot integrate with the large model's inherent knowledge. It's like an ordinary person with a dictionary trying to run a translation company—encountering the same problems that plagued expert systems.

2.7.2 The Harmful Content Problem [?]

In large models, the attention mechanism is correct, but deep learning is flawed. Transformer models based on self-attention, including positional encoding, primarily add Token position information to utilize mutual positional relationships. This is necessary for the attention mechanism to find Tokens' temporal and spatial relationships.

However, through multi-layer deep networks, large models find "optimal coordinate basis clusters" after multiple transformations under error constraints. This "optimal coordinate basis cluster" Token combination no longer matches the original Tokens' temporal and spatial relationships. While some organizational information between Tokens may remain (since deep learning is irreversible, location information is only partially retained), it becomes difficult for humans to understand and exploit. We believe deep learning destroys the original temporal/spatial organization form of Tokens.

Large models perform a lossy translation process, converting human Tokens into their own language. The problem is that humans do not master the large model's language, so we cannot understand knowledge created by large models nor imitate its organizational form to implant "innate knowledge." This is the core problem.

Moreover, because large models cannot achieve few-shot and cumulative learning, they require enormous samples and acquire knowledge shape all at once, further increasing human difficulty in understanding knowledge organization forms. Since machines lack self-needs, they cannot have self-perceived rewards and punishments. Without self-perceived rewards and punishments, they cannot spontaneously create vector projections (information) onto reward/punishment dimensions. In other words, the coordinate basis clusters created by machines lack fundamental dimensions unique to humans: reward, punishment, happiness, sadness, etc.!

Large models' current remedy is RLHF, equivalent to humans adding a reward dimension suffix to particular vectors—adding a reward dimension to the machine's coordinate basis cluster. If training data increases component values in the reward dimension across many diverse, sufficient vectors, this establishes common component combinations' projection to the reward dimension—the machine's reward function. Machines can then predict reward components in output vectors from different decisions (different combinations), preferring outputs with high reward components. This is RLHF's remarkable effect. Knowledge learned through RLHF can generalize. When machines possess their own reward/punishment dimension, they gain preliminary "consciousness" of seeking benefits and avoiding harm, which is why we glimpse a "hazy shadow of consciousness" in current large models.

But this is a patch: machines try, humans score and provide feedback, applicable only in domains permitting extensive trial-and-error. This resembles a PhD graduate lacking "right/wrong" concepts, where parents can only shout "No," "No," "Yes" to instill these concepts, unable to communicate directly except through "Yes" and "No." This learning is inefficient and may perpetually encounter unexpected corner cases.

3.1 Can Big Models Achieve AGI?

We believe large models prove the general direction correct but are not the right path to AGI. In NLP, humans progressed from early bag models and word vectors to EMLO [**?**] until Transformer truly realized the attention mechanism. Combining deep learning with attention mechanisms [**?**] can produce optimized coordinate basis clusters similar to human expression, which is why Transformer exhibits intellectual "emergence."

However, the large model path follows "establish preliminary relationships → adjust coordinate basis cluster → obtain correct relationships under preferred basis cluster." This mechanism requires enormous data and computation, forming knowledge through training that is difficult to update in real time [**?**][**?**]. Meanwhile, the reward function appears after the fact, making it inapplicable to difficult trial-and-error domains like interactive decisions in real environments (autonomous driving, home nannies, industry, agriculture, business, services, government management, etc.).

Additionally, the "task-oriented reinforcement learning" concept is flawed. Humans are "universal" because we face all tasks and decide according to "seeking advantages and avoiding disadvantages." AGI should be the same. With thousands of tasks, task-oriented reinforcement learning can never learn them all, and many tasks have prohibitively high trial-and-error costs.

3.2 What Is the Right Path to AGI?

Current large model problems can be summarized as: (1) The attention mechanism is correct, but deep learning is flawed because it destroys Tokens' original temporal/spatial organization, making generated knowledge difficult to understand and imitate. Without "self-needs," machines cannot have "own ideas" or "independent decisions," leaving them to follow predetermined processes (preset or statistical) passively—inflexible and inadequate for current AI's big problems. (2) The "task-oriented reinforcement learning" concept is wrong. Humans are universal because we face all tasks and decide by seeking advantages and avoiding disadvantages. AGI should be the same. Task-oriented reinforcement learning can never learn thousands of tasks, many with prohibitive trial-and-error costs. No one wants their children used for machine experiments!

Our solution: (1) Implement the attention mechanism without destroying Tokens' original temporal/spatial organization, creating understandable and imitable knowledge. (2) Imitate knowledge organization forms to endow "innate needs." "Innate needs," as a special Token class, form common combinations with other Tokens through attention mechanisms—these combinations are common sense (the world model)! (3) Machines learn only one thing: "how to meet their own needs," handling only one task: "how to meet their own needs." This is general decision-making. (4) By preserving Tokens' original temporal/spatial organization, machines can directly obtain Tokens' temporal and spatial arrangements through linguistic symbols. Since these arrangements are understandable and imitable, machines can directly acquire all experience accumulated throughout human civilization through language learning—no longer needing to retrace the "evolutionary history"!

Step-by-Step Implementation of General AI

Here are the 10 steps to implement our protocol:

Step 1: Tokenize information (like any other AI technology)

Step 2: Matrix the Tokens (build a memory database)

Step 3: Propagate activation values from input Tokens to memory bank Tokens according to similarity relationships

Step 4: All activated Tokens propagate activation values to adjacent Tokens following proximity relationships

Step 5: Each activated Token, in turn, spreads activation values throughout the memory bank

From Step 3 to Step 5, higher similarity yields larger transfer coefficients. Closer storage locations yield larger transfer coefficients. Higher Token memory values indicate larger transfer coefficients.

Step 6: Accumulate activation values obtained for each Token from different propagation paths

Step 7: Resolve activation values of all Tokens over time Steps 3-7 constitute the chain association activation process—the attention mechanism's inference process—where activation values are inference weights.

Step 8: Update each Token's memory value according to its activation value magnitude, with all memory values fading over time Each Token's memory value is its pre-trained weight. In memory, numerous Token combinations exist; those appearing repeatedly contain Tokens that mutually activate, pushing up activation values and obtaining higher memory values. When multiple Token combinations appear in input, they have higher probability of receiving high attention weights. Thus, the chain association activation process is a "Token combination" activation value propagation process.

Step 9: Preset minimum innate requirements (innate knowledge, composed of Tokens + memory value + arrangement). Innate demands represent imitated knowledge organization forms establishing innate knowledge. Innate knowledge can include minimum innate needs, rewards/punishments, emotions, necessary innate safety instincts, and other preset knowledge. This knowledge exists as part of the memory bank, seamlessly integrating with acquired memory to form the overall memory bank. "Fine-tuning" of innate knowledge is achieved through accumulated acquired knowledge (including feedback).

Step 10: Form a fully connected knowledge network Let innate needs, rewards/punishments, and emotions (represented by special Tokens) together with acquired information (ordinary Token information flow) form temporal information flow during machine training and life, stored and processed through chain association activation + attention mechanism to create a fully connected knowledge network (memory bank).

Our scheme results in a memory bank where each Token is a data record consisting of four fields shown in : Record Time, Token, Memory Value, and Activation Value. A large number of Tokens are stored at time intervals, forming a knowledge network through optimization (survival of the fittest via chain association activation + memory and forgetting mechanism). In this network, memory values represent pre-trained weights; activation values represent attention mechanism inference weights. The network contains both objective and subjective Tokens, with attention mechanism connections forming "information." All Token permutations constitute high-dimensional information, while "knowledge" consists of repeatable permutations (including temporal and spatial patterns)—lower-dimensional, more representative, more applicable, and more abstract.

Common sense is further limited to our shared human "knowledge." Our machine memory banks can be inserted, modified, or merged, enabling direct knowledge sharing between machines. For example, a chef robot loading a doctor robot's memory directly acquires medical skills without retraining on combined chef and doctor big data at enormous cost.

4.1.1 Step 1: Tokenize Information

Machines only need to disperse input information according to overall priority and low-resolution priority, extracting underlying Tokens (overall outlines, textures, topology, lines, images, horns, ridges, vertices, voice time-domain/frequency tone, timbre, etc.). Store them chronologically in the memory bank—no identification needed, just save them. Even if Token extraction is random or the algorithm imperfect, our algorithm accumulates common Token combinations (the "world model") to guide machines on "extraction on demand." Common Token combinations contain both common Tokens and their organizational forms.

Thus, Token extraction is a step-by-step optimization process. After Tokens enter the memory bank, their memory and activation values constantly change according to chain associative activation and memory/forgetting mechanisms. Through survival of the fittest, widespread Tokens or Token combinations are retained to form more complex Tokens, while rarely repeated Tokens are eliminated and no longer extracted. Machine Token processing strategy mirrors humans: seek widespread raw data combinations as Tokens—an application of the common information combination first principle, similar to evolution's gift to humanity, as extracting underlying Tokens requires extensive multiplexing for maximum energy utility.

4.1.2 Step 2: Matrix the Tokens

Each Token corresponds to a memory bank record with four fields as shown in . Memory value size indicates memory strength (zero means removal). Activation value size indicates activation strength (zero means inactive). All records spontaneously constitute the entire memory bank through simultaneous storage methods, which include: (2.1) Machines retain Tokens' relative temporal positions in input information. One implementation uses storage space distance to reflect temporal distance—storing Tokens in input order so closer temporal Tokens have closer storage locations. Another method assigns each Token coordinates in memory space containing temporal information. (2.2) Machines retain Tokens' relative spatial positions in input information. One implementation overlays extracted Tokens with original data, preserving spatial relative positions during storage. Another approach extracts overall low-resolution Tokens first, then extracts other local Tokens on demand based on machine decisions. Through proximity storage relationships, local and overall Tokens share both proximity activation and similarity relationships, activating each other and establishing positional connections.

4.1.3 Step 3: Similarity Activation from Input Tokens to Memory Bank Tokens

Each input Token receives uniform initial activation value A0—a preset numerical value adjustable by activation values of activated reward/punishment sym-

bols from the last chain association activation process. The activation values of reward/punishment symbols represent the machine's predicted reward/penalty values for previous input information. Initial activation value A0 affects chain association activation process range. Higher A0 expands propagation because our scheme's activation value propagation coefficient is less than 1. As chain spread increases, propagated values diminish until Tokens obtain activation values below a preset threshold, ending propagation. A0 reflects how much the machine values input information—higher A0 activates more memory Tokens, similar to humans who generate more associations when previous Tokens bring high potential rewards/penalties (e.g., a boss's request).

Similarity activation principles: (1) Higher Token similarity yields larger transfer coefficients (dot product correlation between Tokens). (2) Higher memory values yield larger transfer coefficients (memory values are pre-trained weights). It should be emphasized that the same Token may appear in many memory bank locations with different memory values because the same Token's weight differs by location—similar to attention mechanisms in large models.

4.1.4 Step 4: Proximity Activation of All Activated Tokens

We believe Token proximity relationships represent implicit correlations—closer occurrence times indicate closer potential relationships. These correlations can be counted through chain association activation + memory/forgetting mechanisms. Proximity relationships actually reflect Token combination relationships. If combinations repeat, they become common combinations. We discover common combinations through proximity activation during chain association activation.

Each activated Token transmits activation values to adjacent Tokens: closer temporal positions yield larger transfer coefficients; higher memory values yield larger transfer coefficients. If Tokens in memory banks have close relationships, they were once a combination pattern. High memory values indicate common combinations. If only one Token has high memory value, they are not common combinations. If neither Token's memory value is high, their propagated activation values are low and chain propagation stops quickly, indicating unimportant information with low processing weight. Tokens activate common combinations containing them using "closer temporal position = larger coefficient; higher memory value = larger coefficient," essentially projecting input Tokens onto a coordinate basis composed of Token sets.

If N input Tokens project onto combination X (Token combination) in memory, combination X receives high activation values because each Token activates multiple Tokens in X through both similarity and proximity. Thus, X achieves higher activation values through accumulation. These higher-activation Tokens compose the "model"—the expected model activated by the input vector (N Tokens), the world model. This is essentially a vector decomposition to coordinate basis and an information recognition process.

4.1.5 Step 5: Each Activated Token Spreads Activation Values Through Similarity and Proximity

Each input Token undergoes "similarity activation" and "proximity activation" in the memory bank, with activation value transfer magnitude positively correlated with pre-training weights (memory values). Each activated Token in the memory bank also exhibits positive correlation between "similarity activation," "proximity activation," and activation value transfer magnitude with pre-training weights (memory values). This process continues until all input Tokens complete their "chain activation process."

Thus, beyond activating memory vectors similar to input vectors, machines also activate "antecedents" and "consequences" of similar memory vectors—front and back information in time within the memory bank, possibly across different memory segments, activating different "antecedents" and "consequences." This allows our scheme to speculate on possible former vectors and predict next vectors.

Since our strategy prioritizes "overall low-resolution Tokens," spatial position relationships are established through temporal position relationships. When information enters, machines first extract "overall low-resolution Tokens," store them in the memory bank, then initiate chain associative activation. After completion, decisions are made by counting activation values of activated reward/punishment symbols.

Machine decision-making principles seek advantages and avoid disadvantages. Decisions may involve further information identification or other choices. If deciding to further identify information, machines treat current high-activation Token combination patterns (including language Tokens) as expected models to proactively confirm high-activation Tokens not yet present in input. The method imitates past experience to adjust the sensor system—similar to human pattern recognition. These newly acquired Tokens (e.g., local details) have temporal proximity and partial similarity relationships with original "overall low-resolution Tokens," connecting through mutual activation value transmission. This establishes positional relationships between new and original Tokens. Through memory/forgetting mechanisms, overall low-resolution Tokens and accompanying local Tokens gradually form a "world model."

The world model is not an independent model—it contains Tokens potentially spread throughout the memory bank, temporarily composed of high-activation Tokens motivated by input information. It is not static but distributed, existing temporarily without a separate model in the memory bank.

4.1.6 Step 6: Activation Values Are Accumulated

If activation value propagation paths exist between a Token and multiple input Tokens (directly or indirectly), activation values from inputs accumulate. Thus, Tokens in memory banks associated with multiple input Tokens obtain higher

cumulative activation values from multiple propagation paths. Input Tokens that are mutually associated push up related Tokens' weights in the memory bank—common combinations' activation values rise above the "sea level" of low-activation Tokens. Tokens rising above this sea level constitute one or more "world models." While memories most directly related to input may also obtain high activation values due to short propagation paths, even if not common.

Therefore, our scheme can obtain both "information frameworks" through common Token combinations and attend to specific factual details, making it a "fact database" that can solve GPT's current "hallucination" problem.

4.1.7 Step 7: Activation Values Subside Over Time

All activation values constantly decrease over time. When subsequent Tokens input, related memory Tokens activate. If previously activated Tokens haven't fully subsided, their activation values accumulate. Machine decisions are based on all activated Tokens, considering both earlier and later inputs. This gives machine thinking temporal consistency, solving "omission," "reference," and "metaphor" problems.

Our machine leverages implicit relationships between front and back inputs—this is the attention mechanism! Further, machines adjust initial activation value A0 assigned to input Tokens based on "pros/cons" predicted from the last decision. A0 affects activation value propagation range and accumulation magnitude—adjusting attention intensity according to "pros/cons," surpassing current Transformer technology. This closely resembles human decision-making: a boss prompting you to generate more associations activates more reward/punishment symbols for deeper reward/penalty prediction.

4.1.8 Step 8: Update Pre-training Weight Matrix Through Chain Association

In our protocol, recurring Token combinations likely obtain higher memory values through repetition. Because they are recurring combinations, they mutually push up activation values, obtaining far higher memory values than simple occurrences. Since they repeat, their combinations achieve higher activation values each time, making them easier to activate and gain memory increments—creating a positive cycle. Thus, our machines can learn from themselves, though forgetting pre-existing mindsets is also time-consuming.

In our protocol, machine pre-training statistics are not simple repeatability counting plus memory/forgetting mechanisms, but joint completion through attention mechanism + memory/forgetting mechanism + benefit/harm avoidance principles.

Machine decision-making seeks advantages and avoids disadvantages. In this process, information identification follows benefit/harm patterns. Thus, our

machines build common Token combinations based on their own needs, exhibiting selective recognition of external and internal information.

Memory/forgetting mechanism: All Tokens in memory banks, when activated once, positively update their memory values according to activation value magnitude. Memory values are the pre-trained weight matrix! Since Token permutations cannot be exhaustive, this is an incomplete statistical process, similar to large model pre-training.

Chain association activation process is the attention mechanism's inference process (from local statistical weight to input localization weight calculation) under input Token combination incentives, similar to Transformer's attention mechanism. This process essentially projects input vectors onto coordinate basis clusters established by the attention mechanism. Input vectors can be regarded as original basis clusters formed by impulse functions under input dimensions, while attention mechanism coordinate basis clusters are built on common Token combinations.

The attention mechanism's inference weight matrix is the coefficient matrix for projecting input vectors onto the basis cluster. In our scheme, the chain association activation process, similar to Transformer's multi-layer attention mechanism, is also a coordinate basis cluster projection process: first separate Token projection, then combined projection. After final chain activation, high-activation Tokens' high-weight components form the information "framework." Each framework contains many Tokens difficult to describe specifically, but language symbols, due to high representativeness and repeatability, often achieve highest activation values, becoming representative Tokens of the framework. Thus, in our scheme, activation values are the inference weight matrix.

Both large models and our network are neural networks. The attention mechanism is essentially Bayesian inference—generally, the conditional probability of some Tokens and joint probability of other Tokens, the joint conditional probability of specific Token combinations. This combines Bayesian inference with neural networks. In large models, known probabilities of some Tokens and joint probabilities of other Tokens are determined by weight matrices, performing probability prediction under Token combinations through multiple correlation operations. In our scheme, joint probabilities of some Tokens are explicitly expressed in the memory bank as Token memory values, relative positions, and similarities.

Our attention mechanism realization enables few-shot, cumulative learning with real-time weight matrix updates, making knowledge updated in real time. We don't distinguish between pre-training and inference, enabling lifelong learning. Additionally, our scheme doesn't need the BP algorithm or pre-training, with basic computational load close to large model inference. Therefore, our computational requirements are much less than Transformer, and it can be parallelized, enabling computational localization of the pre-training process. Every machine is a self-training, constantly iterative, evolving agent.

Token extraction can adopt techniques similar to current large models with comparable computational load. The chain association activation process is highly stereotyped and can be implemented at the hardware level with new memory devices, facilitating computational localization and reducing costs.

4.1.9 Step 9: Preset Minimum Innate Requirements

We have realized the attention mechanism, found common Token combinations, and preserved Tokens' original temporal/spatial organization! Therefore, our knowledge network is understandable to humans. We can imitate Token organization forms in the final memory bank to build initial minimum innate memory for machines—equivalent to giving machines minimum innate knowledge similar to human babies.

Innate memory requires minimum "demand systems," "reward/punishment systems," and "emotional systems." The method uses special Tokens to represent each "demand," "reward/punishment," and "emotion," then imitates pretraining forms (appropriate Token arrangements + appropriate memory values) to implant minimum innate knowledge.

In daily life, let these special Tokens representing "demand," "reward/punishment," and "emotion" train together with external triggering Tokens, activating chain associations and undergoing memory/forgetting together. Through the attention mechanism, these special Tokens establish common Token combinations like other Tokens.

We must preset machines' minimum "demand system," "reward/punishment system," and "emotional system" so external world Tokens (including machine state parameters) can trigger these special Tokens to establish information flow. Through chain association activation + benefit/harm avoidance decisions + memory/forgetting mechanisms, machines gradually obtain the most common and concerning Token combinations.

Thus, we establish connections between "objective world common Token combinations" and "needs." "Objective world common Token combinations" are "objective common sense," while combinations of "objective world common Token combinations" and "demands" are "subjective common sense." "Objective common sense" and "subjective common sense" constitute "common sense."

Common sense is the "world model," containing both human cognition of the external world and relationships between the world model and "self" established by humans. Tokens include not only static features but also simple dynamic features (rotation, swing, etc.), so the world model is not static or fixed but created under input Token excitation! Each world model differs, directly related to its experience. In our scheme, the machine's "world model" relates directly to training data and life experience!

With the world model, input Tokens can activate reward/punishment Tokens, emotion Tokens, and demand Tokens through chain association activation. The

activation value transfer path from input Tokens to these feature Tokens is a logical reasoning process compatible with neural networks—explicit, understandable, and imitable, making machine decisions visible.

Step 9 is essentially the first step in actual general AI creation. We can train experimental data through previous steps to obtain and understand machine-created knowledge organization forms, then imitate these forms to implement Step 9.

(1) Preset a basic needs pros/cons system related to machine life activities. For example, give battery data a reasonable interval, preset a "hungry" symbol in "innate memory," place "punishment" and emotion symbols representing "hunger" next to the "hungry" symbol, and give them appropriate memory values. When power is insufficient, the vital state monitoring program gives initial activation value to the "hungry" symbol in innate memory. Its activation value spreads chain-wise throughout the memory bank, activating the "hungry" emotion symbol and "punishment" symbol, giving the machine a "hungry" mood and "punishment value." To avoid the penalty value, the machine uses its experience to actively find a plug to charge!

(2) Preset "higher-order needs" values for machines, requiring the simplest communication means and cultivated values. Values must be cultivated from childhood, so we need to educate robots about "values" from an early age. Education requires "reward" and "punishment," so machines must recognize these from the start to initiate the first learning step!

Thus, we must imitate acquired memory network organization forms to give machines innate knowledge capable of recognizing the simplest "reward" and "punishment." For example: preset basic head-nodding features (assuming X Tokens) and head-shaking features (assuming Y features) without needing precision. Place a "respected" symbol and "reward" symbol next to nodding Tokens, give them higher memory values, and make their relationship long-term memory. When partial nod Tokens appear in input, the machine obtains "reward value" through chain association activation. In pursuit of "reward," the machine may plan various future decisions to obtain human nods!

Similar to children starting from the simplest communication methods to gradually acquire complex learning abilities, they establish a "reward function" logical chain: "milk," "pacifier," "bottle," "milk powder can" → … → "academic performance," "house, car" → "social status" → "worldly ideals."

After training, the machine's memory bank contains numerous reward/punishment-related Token symbols, with Token combinations closely related to these reward/punishment Tokens forming causal relationships. These combinations represent values. Any machine value can be established by presetting innate communication means and step-by-step cultivation. In fact, humans are the same—no one is born a "saint."

[Figure 1: see original paper] shows a schematic diagram of establishing "innate minimum requirements."

4.1.10 Step 10: Form a Fully Connected Knowledge Network

Our scheme results in a network where each Token comprises four fields: time mark, Token itself, memory value, and activation value. Numerous Tokens are stored according to time intervals, forming a knowledge network through optimization (survival of the fittest via chain association activation + memory/forgetting mechanism). Memory values represent pre-trained weights; activation values represent attention mechanism inference weights.

Our network contains both objective and subjective Tokens, with attention mechanism connections forming knowledge, among which common knowledge is "common sense." This enables machines to predict pros/cons and make independent decisions because they have "demands" and "logical chains" related to demands (activation value transfer links formed by Tokens). Driven by demands, they actively learn and iterate themselves—for example, going to recharge or visiting the library!

In our scheme, knowledge and decisions develop around "demand," which is the core reason our machines achieve "universality." They face only one task: "needs," rather than various "external tasks." Our solution is "active intelligence," while others are "passive" intelligence.

Our scheme enables few-shot learning, real-time knowledge updates, and integrated training/usage processes, making machines lifelong learners and self-iterative. Since machine knowledge exists as memory banks stored chronologically, with memory values gradually optimized on the original memory bank, different memory banks can be directly stitched into larger memory banks. Combining a chef's memory bank with a doctor's memory bank gives robots both skill sets without retraining on combined chef and doctor data—a capability current AI technology cannot achieve. Large models must train on massive amounts of both doctor and chef data for a machine to master both skills, making it a wild hope for machines to have "all kinds of" abilities through this training method.

4.2 Example of Memory Value and Activation Value Changes

[Figure 2: see original paper] shows a simple example of memory value and activation value changes during associative activation. For simplification, assume the machine's memory bank is empty and receives input for the first time (without preset innate memory). Suppose from time t0 to t7, the machine input Tokens are "We hope for world peace." In actual processes, machines should adjust initial activation values for all input Tokens according to currently activated reward/punishment symbol values. Here, lacking a value system, we

assume default input Token initial activation value of 90 (assuming activation value range 0-255). After chain association activation, assuming the memory curve, Token activation value is 90, current memory value is 0, and memory value increment obtained is 126: dm = f(m0, A0), where m0 is current memory value and A0 is current activation value. Memory value update increments are positively correlated with activation values. All memory and activation values decrease over time (exaggerated descending gradient employed here).

From time t9 to t19, the machine receives second input Tokens: "Peace makes our world better." Through "similarity activation," the first Token "and" activates Token "and" in the memory bank, giving it activation value. Because "and" in memory bank has high memory value, it obtains activation value from initial activation value with large transfer coefficient. Similarity activation transfer coefficient T = f(S, m0), where S is similarity (Token vector dot product) and m0 is transmitted Token's memory value. Activation value transfer coefficient is positively correlated with similarity and memory value.

Simultaneously, "and" Token in memory bank initiates chain propagation because its activation value exceeds the preset threshold. In chain propagation, it first activates near-relationship Tokens "flat" and "bound" through "proximity activation." Proximity activation transfer coefficient T = f(D, m0), where D is temporal distance between two Tokens and m0 is transmitted Token's memory value. Proximity activation value transfer coefficient is anti-correlated with temporal distance and positively correlated with transmitted Token's memory value.

After "flat" and "bound" obtain activation values, if exceeding the preset threshold, they also initiate chain propagation, searching for similar Tokens in memory library for activation value propagation and propagating to adjacent Tokens. Both processes' transfer coefficients are positively correlated with memory values.

Through chain association activation, input Tokens can activate the entire memory repertoire and associated Token combinations. Activation range depends on initial activation values, adjusted by value prediction.

After completing chain association activation for the second input, we see that in memory bank Tokens, the "peace" Token combination has the highest memory value and approaches, so each Token obtains higher activation values due to high memory value. Meanwhile, "and" and "flat" not only have chances to obtain high activation values themselves but also pass activation values to each other (proximity activation). Due to their high memory values and high transfer coefficients, through activation value accumulation, they become a Token combination easily obtaining high activation value weights.

Second, the "world" Token combination in memory bank, similar to "peace" Token combination, obtains the second-highest memory value, also becoming a Token combination prone to high activation value weights.

In [Figure 2: see original paper], we use Chinese characters to represent tokens, though word vectors could also represent tokens without process differences.

Thus, only two sentences establish relative Token "weights." Machines can build correct common Token combinations and their memory values. This memory value corresponds to combination "common degree"—actually the local statistical value of combination occurrence probability from training data. Activation value is the dot product process (projection) of input Token combinations onto "common Token combinations" (basis cluster) based on this local statistical value.

We use a human-like learning method that is highly efficient, enabling few-shot, cumulative, real-time learning. It doesn't modify "old knowledge" parameters, avoiding "catastrophic forgetting." It doesn't require BP gradient optimization, so computational load is consistent with large model inference.

5. We Implement the Three Conditions Proposed by Professor Yann LeCun

With the three deep learning giants and Turing Award winners, Professor Yann LeCun believes AGI's right direction is the "world model" through "humanoid AI." They propose three conditions: (1) Need a world model, including demand modules modeling basic needs like happiness and hunger, and value modules predicting value. (2) Need logical reasoning ability compatible with neural networks (current reasoning relies on plug-in symbolic reasoning). (3) Need "general decision-making ability" that can decompose decisions top-down, not requiring intensive training a million times per task!

Although they propose these ideas, they have no complete technical solution. Our scheme achieves all three conditions.

5.1 We Built the World Model

Input Tokens activate Token combinations in memory; high-activation Token combinations are the activated "world model" (some Tokens may already appear in input, others not yet). According to "seeking benefits and avoiding harm" predicted decision process, machines decide whether to further confirm existence of other "high-activation Tokens"—this is "pattern recognition." The world model is "common sense": Token combination patterns composed of subjective Tokens (demand, reward/punishment, emotion) and objective Tokens. Humans use "common sense" for "pattern recognition."

After each new information input, machines conduct chain association activation, then store Tokens in "simultaneous storage" mode reflecting time intervals between Tokens (e.g., closer Tokens in time have closer storage locations or time information). Every time machines obtain new Tokens, they need updated activation values to find ways to achieve reward and avoid punishment. These path

sets form the overall response path—a network-like structure where many local paths may lead to both reward and punishment.

Because activation value transfer paths to reward symbols (or punishment symbols) realize advance and stepwise reward/penalty functions, we solve the sparse and lagging reward function problem in current reinforcement learning. Machines can find initial optimal response paths through AlphaGo-like search processes.

If overall reward/penalty values don't reach acceptable preset values (or don't converge), machines cannot decide whether to select or exclude specific paths to maximize benefits. Machines need further information identification, adding more Tokens to subdivide certain reward/penalty activation value transfer paths to help select or exclude specific paths. This is the process where machines spontaneously and actively find information to help make decisions, iterating until reward/penalty statistics reach accepted preset values or converge.

When further identifying information, high-activation Tokens either have high memory values (representative Tokens of thing classes) or are closely related to input Tokens (similar or frequently near). Therefore, high-activation Token combinations activated in memory are representative Token combinations related to input information. These representative combinations are the "world model" temporarily created by machines—the "expectation model." It is both a summary of past experience (Tokens memory values after survival of the fittest) and directly related to current specific input, temporarily created through high activation values—the machine's "expected model" for current input Token combinations.

The spatial or temporal relationship between Tokens in the machine's expected model, between already-present and not-yet-present Tokens, given Tokens' temporal and spatial locations, predicts locations of not-yet-appeared Tokens. These high-activation Tokens not yet appearing in the expected model are expected Tokens. Machines assign sensor search temporal/spatial locations based on expected Tokens' time, space, and size in the expected model, determine sensor type based on expected Token properties (speech, image, touch), and determine resolution based on expected Token properties (size). This is the machine's "on-demand identification" process, performable iteratively.

Selective attention extracts Tokens from input information according to recognition intervals and resolutions. This solves wireless granularity of image information (machines extract image information on demand). When extracting specific interval data, machines preferentially extract overall topology, shape outlines, main lines, and main texture Tokens in the selected interval using overall-feature-first approach.

Machines then obtain relevant memory through chain association activation, combining these memories into different weight expectation models according to weights. Using decision processes, machines determine whether to further identify input information based on activated reward/punishment Tokens (their

activation values are expected reward/penalty values) or respond to input information.

If machines decide to further identify input information, they extract "expected Tokens" from input by imitating past experience of obtaining "expected Tokens." Thus, machines constantly iteratively extract Tokens from input information through attention mechanisms, possibly using different sensors, resolutions, and recognition intervals for each extraction. For the same input, machines may extract different types, intervals, and resolutions of Tokens, using these combinations to form a "hierarchical representation" of the same thing—extracting information step-by-step in overall low-resolution intervals over time.

High-activation Tokens form expected models based on two sources: common features of similar things (widely found, highly repetitive, usually high-memory-value Tokens) and specific Tokens directly activated by similarity in input Tokens. The former enables machines to first identify large categories (abstract concepts) through common features, then gradually add more Tokens to limit scope (abstract to concrete concepts). The latter, due to short activation paths, quickly locates expected models through special Tokens.

Thus, input information identification identifies large categories through common features, then determines specific subcategories through unique features. Machines iteratively increase identification Tokens through selective attention. Previously activated Tokens' activation values fade over time; if re-activated by new input Tokens, values are maintained; if unrelated to new input, values fade and gradually exit decision processes.

The "world model" contains two aspects: (1) Machines iteratively know the world through "pattern recognition." (2) Machines know the world through "pros/cons" because "pros/cons value" is the core world model guiding all human behavior. Thus, we implement the "world model."

5.2 We Achieve Logical Reasoning Compatible with Neural Networks

All "reward/punishment" Tokens motivated by input Tokens have activation value sizes as value predictions. The propagation path from input Tokens to activated reward/punishment Tokens is reasoning fully compatible with connectionism! A memory network is a neural network where Tokens transfer according to activation values. Activation value transfer's essence is the inference process realizing the attention mechanism.

Token combinations, through chain association activation, activate their common Token combinations. Through activation value accumulation, they can realize from input combination (known N Tokens' specific combination probability) to most relevant Token combinations (M Tokens' specific combination probability), with final activation value distribution in memory library as Bayesian inference results.

Large models' attention mechanisms have achieved logical reasoning compatible with neural networks, but with two defects: (1) Deep learning destroys original Token temporal/spatial organization, making knowledge difficult to understand and imitate. (2) Lack of "subjective Tokens" (needs, emotions, pros/cons) makes big model inference flawed.

Turing Award winner Professor Yoshua Bengio believes general AI's most important step is combining neural networks with causal reasoning. Our scheme achieves this: the memory network is fully connected, from input Tokens to activated "world model" is causal reasoning of objective world organization; from input Tokens to activated "subjective Token combinations" (representing demand, emotion, reward/punishment) is causal reasoning between objective world and machine's own needs.

Thus, we implement "combining neural networks and causal inference." Current large models have realized objective reasoning and some subjective reasoning, but their reasoning processes are difficult for humans to understand and imitate, making them hard to use.

5.3 We Achieve Hierarchical "General Decision-Making Capability"

Machines perform reinforcement learning on only one task: "How to meet their own needs?" and handle only one task: "how to meet their own needs." Thus, our machines' decisions "face their own needs," while other AI solutions face various "tasks themselves." Information input produces various associations—good and bad. Reduce probability of Tokens bringing "punishment," increase probability of Tokens bringing "reward." This is "general decision-making," similar to human decision-making—universal!

With advance and stepwise reward functions, machines have "decision-making ability." With the general goal of "seeking advantages and avoiding disadvantages," machines achieve "general decision-making" capabilities.

5.3.1 Current "Machine Learning" Is Not Real "Machine Learning"

Facing new tasks, humans predict "good/bad" of different decisions based on experience, trying at most a few schemes. Current machines rely on reinforcement learning to "keep trying," either (1) trying a million times to see results (Google AI for games) or (2) having humans tell them good/bad (GPT-4, large models, RLHF) [**?**], then gaining decision knowledge.

Thus, current machine learning uses "try first + then eliminate"—it should be called "machine evolution," not "machine learning." We propose AGI requires real "machine learning." Real machine learning should, like humans, predict "good/bad" along different decision paths based on past experience when facing new tasks, trying limited solutions to obtain decision knowledge. Furthermore, real learning should, like children's learning, directly acquire accu-

mulated human experience through language. Facing new tasks, attempts are unnecessary—direct success! For example, in laboratories, teachers pass existing human decision-making experience to children through language teaching. After obtaining knowledge, children can use language-acquired experience to directly complete experiments through environmental interaction, even if it's their first time.

Real tasks and scenes vary greatly; humans cannot reinforcement-learn each task type in numerous scenes. We must change thinking: turn all tasks into one task—"How do you meet your needs?" All machine training trains this task. Facing this task, machines have abundant "states" and "policy knowledge," enabling prediction of potential "pros/cons" estimates for "different decisions."

Tasks given to machines are background information for the "how to meet their needs" task. If "gaining human recognition" is a machine need, machines incorporate "completing human tasks" into overall pros/cons statistics while pursuing "meeting their own needs." This resembles humans facing bosses, weighing pros/cons to make different decisions. One decision may be proactively finding more information to analyze task pros/cons before deciding. Actively seeking more information becomes a new self-assigned task. If machines make the same decision, they assign themselves tasks—machine self-programming. Our machines use this decision-making process: weighing pros/cons with potential to actively find information maximizing benefits.

5.3.2 How to Achieve Real "Machine Learning"?

Ten years ago, we believed creating real "knowledge" should start from information statistics perspective. Unlike deep learning, we thought machines should learn according to human models—few-shot and cumulative. We initially tried "symbolic expression," "causal logic," and "knowledge network."

After years of attempts, the first step was blocked. How to express "dog" symbolically? Need to pick all "dog" characteristics. But "dog" can be animal or person, "celebrated character" or "despised character"—"dog" meaning varies greatly by context. "Dog's" essence is the sum of relationships between "dog" and all other things. "Dog" must be placed in the whole knowledge network, defined by relationships to all other knowledge. Thus, "symbolism" doesn't work because "dogs" cannot be separated from other knowledge! Our first conclusion: need a "fully connected knowledge network" similar to deep learning.

Because "dog" must be defined by relationships to all other knowledge, sufficient knowledge is needed to understand what a dog is. Knowledge volume must be sufficient to understand through enough background knowledge. This is our second conclusion.

Looking back, isn't this what large models do? "Deep learning" creates fully connected networks, and large models use massive knowledge to build fully connected knowledge networks. So why don't we see robots on streets? Because

knowledge networks alone are insufficient—machines must also "interact with the environment to make decisions!" Research shows humans make over 30,000 decisions daily. Beyond industry, only reinforcement learning algorithms enable machines to make their own decisions.

One possible AGI path is: large model + reinforcement learning algorithms. GPT-4 already achieves "full knowledge + fully connected network + RLHF" (RLHF is reinforcement learning). Google published the Gato model in 2022, taking the "all knowledge + fully connected network + reinforcement learning" path.

So why don't we see Google robots on streets? The core obstacle is reinforcement learning algorithms requiring two prerequisites [**?**]: (1) Machines need to know reward information obtainable at different decision paths. Since reward information is scarce and delayed, this requires extensive trial-and-error training. (2) Machines need to search all possible decisions.

These conditions are perfectly satisfied in games—constant trial-and-error possible, decision search space bounded (and prunable). But real life has many problems without constant trial-and-error (e.g., childcare—no one wants endless experiments!) and no clear boundaries, making solution impossible! That's why Google keeps launching complex strategy game AIs but cannot launch basic "home nanny robots." In daily life, most decisions are far less complex than game decisions, but real life prohibits massive trial-and-error and lacks clear information boundaries. These two difficulties prevent OpenAI or Google from launching "big model + reinforcement learning" beyond massive trial-and-error domains. AIGC remains far from AGI!

Our decision-making plan is also essentially reinforcement learning, but only on how to seek advantages and avoid disadvantages. We use chain association activation to automatically limit search range—searching only activated information! Moreover, we use the logical chain of "Tokens" and "reward/punishment symbols" to automatically predict reward/punishment information, rather than obtaining it only through post-hoc feedback. Thus, we perfectly solve the problem that Google's decision-making AI can only play games! This is because we simultaneously realize "objective common sense" + "subjective common sense." Current technical routes realize "objective common sense" first, then establish "subjective common sense" through "RLHF," obtaining "subjective common sense" through post-hoc feedback, applicable only to massive trial-and-error domains.

5.3.3 Implementation Process of "Universal Decision-Making"

Machines in any environment receive input information including all sensor information. At any moment, environmental information is part of input information. Machine-environment interactive decision-making includes two intertwined, parallel aspects: (1) Optimal decision selection. (2) Decision execution process.

The first question "universal decision-making" must solve is: What is the reward function? Inside GPT-4 and AlphaGo, rewards come from final external feedback. In our AGI, rewards come from "reward" and "punishment" symbols activated by external information, with sizes as their activation values.

Step 1: What is the purpose? When information inputs (external + machine monitoring information), some reward/punishment symbols activate. Each transfer path from input reward/punishment symbol activation values is a potential logical link for generating reward or punishment. If each underlying feature truly realizes on this logical link, then reward/punishment spread by this link also realizes. Therefore, machine response to any input information is the same: increase probability of reward logical link occurrence, reduce probability of punishment logical link occurrence, achieving benefit-seeking and harm-avoidance.

Step 2: How to plan with purpose?

1. How to increase reward links and reduce punishment links? Increase or decrease realization probability of high-activation Token combinations on the link. When high-activation Token combinations on the link are true, activation value spread along the link is true, making final activated reward or punishment true.

2. How to operate specifically? From input information reward/punishment symbol activation value transfer paths, select N Tokens with highest activation values—the top implementation paths causing reward or bringing real implementation. Machine goals: (1) Let Tokens on reward path implement (imitate past experience to make them appear in input information). (2) Make Tokens on punishment path impossible (imitate past experience to avoid them appearing in input information).

Thus, from input reward/punishment logical pathways, select N highest-activation Tokens containing activation value propagation paths as top implementation paths. Why select only N highest-activation Tokens? Because these Tokens are either representative (high memory values) or closely related to input Tokens (good match). Due to small numbers and fewer attribute restrictions, concepts most closely related are usually "abstract concepts." Frequently used language Tokens often obtain high activation values, becoming core Tokens constituting "abstract concept" combinations, making language symbols concept representatives (e.g., "eating," "escape"). Note that "abstract concepts" are not exclusive to language—animals also have "top-level decisions."

Thus, machine decision-making prioritizes "abstract concepts," then gradually adds more Tokens to form more concrete concept combinations—a top-down, gradual decision-making and execution process we call "segmented imitation."

Specific segmented imitation example: Consider input Token set A and response Token set B. Machines find high-activation Tokens through A and B chain association activation. These Tokens connect to both A and B, obtaining

activation values from both, becoming bridge Tokens linking A and B. This iterative process enables top-down, layer-by-layer decisions.

Computer implementation: (1) External input Tokens determine reward/punishment symbol activation values (Tokens exceeding preset values become targets), establishing level targets. (2) Starting from highest-activation reward/punishment symbols, find N highest-activation Tokens on activation value transfer paths from input to each first-level target—these are logical links to realize corresponding reward/punishment. Tokens on links are secondary targets. (3) Machines treat each secondary target as new target, as new input Tokens, give them initial activation values, and initiate chain association activation again. Tokens with highest activation values are Token combinations associated with both external input Tokens and secondary target Tokens. Due to activation value accumulation and extinction, only Tokens associated with most recent input Tokens maintain activation state. These become level-3 goals. (4) This iterative process breaks down each level goal into hierarchical logical links. (5) Each decision expansion selects different reward/punishment values for accumulation. According to benefit/harm principles, machines choose subpaths bringing reward values and avoid those bringing penalties, increasing cumulative reward value. When total reward/penalty value converges (cannot be further improved), benefit is maximized. Machines stop expanding and enter execution.

Why select only N highest-activation Tokens per expansion? Because past experience cannot fully match current reality. Selecting only highest-activation Tokens means the "model" is either abstract (widely applicable) or closely related to input Tokens (good match). Selecting only N highest-activation Tokens achieves empirical generalization automatically.

Example: Machines have hammer-nail experience. Needing to hit nails without a hammer, with input Token "stone," to achieve primary goal (reward symbol or punishment symbol—complete task, reward, or avoid punishment), activated logical links may contain hammer-representing Token combinations. These become secondary targets.

Through chain memory association activation, machines may find M hammer target activation value transfer paths—from "memory toolbox" or "borrow from teammate experience." Activation value transfer paths are secondary reward paths. Since stone-related Tokens appear in input, total Tokens of hammer and stone (weight data, size, hardness sensation, etc.) likely obtain higher cumulative activation values, selectable as first N high-activation values. They become bridge Tokens, making stone-related Tokens a secondary reward path. This is empirical generalization: through shared Tokens between stone and hammer, stone Tokens transfer activation values to reward symbols because "stone" and "nail hammer" share some common attributes (Tokens repeatable across various "hitting nails with nail hammer" scenes), forming the bridge of experience generalization.

Which secondary path to rewards does the machine take? Machines activate Tokens space updated according to new chain association activation process and reselect decision paths based on benefit/harm principles. Some paths may bring both rewards and punishment, making reward/penalty calculation difficult. If reward/penalty statistics don't converge, machines decide to further identify information to converge each reward/penalty transfer path.

For example, "toolbox in memory" needs to confirm current probability of "toolbox" appearing in input (implementation). This probability can further converge this path's reward/penalty value. Confirming "toolbox" presence becomes a new self-created goal. To achieve this, machines imitate past experience. If past memory shows toolbox hanging at waist, the most likely imitation process is using "hand" to pat waist, reproducing past hand-touching-toolbox sensor data combinations. This decision costs least power and time while maximizing machine benefits—preferred decision path.

Another example: "borrow from teammate" path requires increasing implementation probability of Tokens on this path to transfer greater activation value (greater reward) to reward symbol. Most likely imitation experience is looking around or asking.

Thus, machine decisions are complex, potentially nesting W decision/execution processes, but always with the sole goal of "seeking advantages and avoiding disadvantages." All decisions derive from this goal, making machine decisions very flexible, always changing according to environmental states, with no preset processes except "Seek the best and avoid the bad."

Step 3: With planning, how to implement? Execution improves by mimicking past experience or reducing Token probabilities: (1) Choose a small number of highest-activation underlying features to abstract decision path. (2) Add more high-activation underlying features to embody abstract decision path. (3) Iteratively repeat steps 1-2 until decisions decompose to executable drive commands (send waveform to speaker, drive command to motor, display data to screen, set parameters to facial expression system, etc.). (4) New input information may appear anytime, changing memory bank activation values and reward/penalty situations, so machines may change original plans anytime during optimal response path implementation.

Step 4: Segmented imitation during decision-making and execution. Machines find experience related to current input through chain association activation. Among these experiences, a small number of high-activation Tokens are usually representative abstractions. These experiences contain "antecedents" and "consequences" related to input Tokens—objects of empirical generalization.

Empirical generalization's essence uses existing process effects to achieve unfinished process effects. In our scheme, this is automatically completed by activation value transfer of "common Tokens" in the two processes. Since Tokens are inconsistent between processes, this corresponds to mismatch problems in

empirical generalization, but our scheme automatically completes generalization through common Tokens' activation value transfer.

Notably, machine concepts are formed by various Tokens through large stereoscopic networks. The same Tokens may distribute across different memory segments, likely coming from both own experiences and linguistic symbol inputs. When language symbols activate, related Tokens they represent also activate. Language symbol sequences contain expression Token combination sequences, so behind language symbol sequences are Token temporal/spatial information flows. Their temporal/spatial combination order is "causal relationship," forming close activation value transfer relationships through language symbols' chain association activation process. This close activation value transfer relationship itself is "experience." Thus, in our scheme, experience comes not only from machines themselves but also from "others' experience" gained through linguistic symbols. Our machines can both learn "experience" through language symbols and imitate "experience" through Token information flows composed of language symbols.

6. Our Scheme vs. Current Large Model Approaches

Our solution solves the following problems:

6.1 How to "Build Up Common Sense"

Deep learning destroys original Tokens' temporal/spatial relationships! Our scheme uses "chain association activation + memory/forgetting mechanism" to realize attention mechanisms without deep learning, preserving original Tokens' temporal/spatial relationships. Original Token combination patterns are exactly the basis of human "concepts." Thus, our "knowledge" is understandable and imitable by humans.

In our scheme, "knowledge" is essentially Token permutation relationships in time and space—predictions of different Token permutations by the agent. Token temporal/spatial arrangements are inherently "causal," not simple proximity but repeatable relationships spanning potentially large time/space gaps. Through chain association activation, large-span Tokens form close activation value transfer relationships—this is knowledge. If knowledge includes Tokens related to "needs," "emotions," and "pros/cons," it can predict potential pros/cons, making Token arrangements represent "knowledge." Common permutations are "common sense."

6.2 The Question of Machine Consciousness

We solved how to give machines "self-needs." Therefore, machines can make independent decisions, self-evolve, have emotions, and pursue "self-needs," making our machines "conscious."

6.3 The "Universal Decision-Making" Problem

Facing any task, machines decide according to "seeking good and avoiding harm." Human-assigned tasks are by-products of machines pursuing "self-needs." This is identical to completing boss-assigned tasks while pursuing "self needs." If conflicts arise, you make flexible decisions according to benefit/harm principles, testing the boss's true intentions and bottom line—making decisions very flexible!

6.4 The "Language Understanding" Problem

Because we don't break original Token temporal/spatial relationships, Token temporal/spatial sequences represented by language sequences can be understood and imitated. Machines can learn various skills directly through language, like humans reading oven manuals to start toasting [**?**][**?**][**?**].

6.5 Advantages

6.5.1 Advantage 1: Handle tasks prohibiting extensive trial-and-error
Such as autonomous driving, home nannies, elderly care, child companionship, and "workers, peasants, soldiers, business" roles. As "humanoid" AI with general decision-making and language skill learning, current big models cannot handle these!

6.5.2 Advantage 2: Solve the "hallucination" problem
Large models have only locally statistical "common words" without factual memory. Our scheme first stores memory then extracts common information, possessing integrated "fact database" and knowledge.

6.5.3 Advantage 3: Learn and imitate skills directly through language
Preserving Token combination temporal/spatial relationships makes spatiotemporal relationships represented by language understandable and imitable! No current or future large model can achieve this. For example, on day one at a bakery, it asks the boss for an "oven" manual, reads it, and starts baking without individual training!

6.5.4 Advantage 4: Safer
(1) Current AI is single-goal, "one-minded" in decision-making, a black box ignoring everything outside the goal. How dangerous if "stuffy pot" + "one-track mind" AI controls your life! Such AI may bring incalculable disasters due to wrong understanding. (2) In our scheme, machine "demand types" can be preset, values trained, and human values aligned. Machines always consider various goals, avoiding "extreme" behavior. Moreover, decisions are visible, modifiable, "white box."

7.1 The Chain Association Activation Process as the Attention Mechanism

We posit that the essence of knowledge is information, and that human knowledge represents only a minuscule subset of all information—constrained by our limited resolution capabilities. For instance, the precise spatiotemporal arrangement of atoms A and B on a blade of grass constitutes information, yet it remains beyond our perceptual recognition. Through evolution, humans have developed the ability to recognize Tokens, which represent the smallest units of information commonly utilized by our cognition (e.g., the concept of a straight line). Tokens themselves function as "world models"—the fundamental building blocks with which humans construct the edifice of knowledge. This evolutionary endowment of pattern recognition, employing models such as Tokens to interpret surrounding information, dramatically enhances the energy efficiency of information processing.

If we were to chronologically arrange the Tokens representing all phenomena from the "Big Bang" to the present moment according to their spatiotemporal relationships, we would obtain an information tensor encompassing the entirety of human knowledge. Should external agents wish to comprehend this knowledge repository, they would necessarily perform statistical analyses on these Tokens. Three fundamental questions arise: First, how many independent Tokens exist? In our framework, similarity relationships address this query. Second, what is the distributional frequency of each Token? Repetitive relationships provide this answer. Third, how are these Tokens arranged? Proximity relationships resolve this question. Thus, through the chain association activation process, coupled with memory and forgetting mechanisms, our approach essentially constitutes a statistical description of information.

In the attention mechanisms of large-scale models, the correlation between Token combinations is inferred through pairwise Token correlations, which are then iteratively compounded to deduce higher-order correlations. This iterative process yields the interrelationships among diverse Token combinations. The pre-training phase discovers the "optimal coordinate basis" through trial-and-error methods (deep learning), which—leveraging the attention mechanism—captures only "common information." The fundamental nature of this process is Bayesian inference: determining the conditional probability of specific Tokens based on partially known probabilities.

In our scheme, Token correlations are acquired through induction. The chain-type associative activation process utilizes correlations obtained during pre-training (partially known probabilities) to derive the conditional probability of a particular Token's occurrence. While the chain association activation process discovers correlations (emulating the reasoning mechanism of attention), the memory and forgetting mechanisms perform induction.

7.2 The Core of the Attention Mechanism: Creating "Common Sense"

Knowledge represents the arrangement of Tokens, whereas common sense constitutes the conventional patterns of Token arrangement [16]. A fundamental challenge for large models lies in converting human knowledge (Token arrangements) into their own knowledge system—a transformation that deep learning disrupts by destroying the original spatiotemporal relationships of Tokens, rendering machine knowledge incomprehensible and inimitable to humans. The machine subsequently employs this alien knowledge system to solve problems, then translates the solutions back to human-understandable form. From the machine's perspective, the organization of Tokens is preserved because it successfully identifies "common information." However, from a human perspective, the knowledge generated by machines cannot directly interconnect with or be borrowed from human-created knowledge.

Because the underlying languages of these two systems cannot communicate directly, imparting "innate knowledge" to machines—such as innate requirements, reward-punishment functions, or emotional mechanisms—proves exceedingly difficult. Consequently, we must resort to remedial measures like RLHF or plug-in knowledge bases, enabling only binary "yes/no" communication. This limitation constrains robots to being "scripted" and "nerdy," incapable of genuinely flexible problem-solving.

Therefore, the cornerstone of our scheme is establishing "common sense" without disrupting the original spatiotemporal organization of Tokens, while necessarily incorporating the machine's own "subjective common sense."

To achieve this, we employ information Tokens that preserve spatiotemporal information during storage, implement a chain association activation process, and utilize memory and forgetting mechanisms to facilitate the inductive transfer of activation values between Tokens. Simultaneously, we emulate the organizational structure of "common sense" by preconfiguring Token combinations that represent innate demands, reward-punishment functions, and emotional mechanisms. We then enable the machine to make autonomous decisions and self-evolve according to the principle of seeking benefits while avoiding harm, continuously expanding its memory repository around innate knowledge to form a comprehensive knowledge network. This process ultimately engenders both "objective common sense" and "subjective common sense."

7.3 Our Singular Capability: "Creating Common Sense"

To "create common sense," we must first address the challenge of (1) endowing machines with self-needs. Solving (1) necessitates solving (2) how to create comprehensible knowledge. Solving (2) requires addressing the problem of constructing a fully connected knowledge network without deep learning, thereby enabling subjective and objective Tokens through the attention mechanism—

this constitutes common sense.

Establishing relationships between subjective and objective Tokens via the "front" and "step" components of the excitation function can realize "general decision-making capability" through the principle of "seeking advantages and avoiding disadvantages." Driven by "self-needs," machines can thereby achieve "self-evolution."

7.4 Establishing an Infant AI

The concept of "building a baby machine that learns lifelong and grows itself" has existed for years, but our team is the first to propose detailed implementation steps.

8 A Simple Example

We illustrate the machine's decision-making and response process through the following scenario: Lao Wang is on vacation in another city with his assistant robot, staying in a hotel room.

When Lao Wang says "Hello…", numerous Tokens activate in the memory bank. However, among these activated Tokens, no reward symbol exceeds the preset threshold A1, and no punishment symbol exceeds threshold P1. The robot continuously receives external and internal sensor information, extracting Tokens from this data at low resolution and storing them in the memory bank. Following the same procedure, these Tokens receive relatively low initial activation values. Consequently, during the subsequent chain association activation process, the propagation range of activation values remains minimal, and the chain activation completes rapidly.

The machine then updates memory values. Because the activated Tokens obtain low activation values (due to both low initial values and limited propagation), their memory value increments are small, and much of their information is forgotten within a short period. Simultaneously, since the activation values of reward and punishment symbols in the memory bank remain relatively low—indicating minimal potential rewards or punishments—the optimal decision path formed by the machine is to continue receiving information. This is because expending power itself constitutes a punishment; without obtaining rewards, the optimal decision is to avoid wasting energy.

Following each chain association activation process, the machine checks whether any reward or punishment symbols exceed the preset thresholds. In this case, the optimal response is to extract Tokens from the incoming information at low resolution and store them in the memory bank, perpetuating the aforementioned cyclic process.

Suddenly, the audio processing system introduces a series of audio Tokens (still extracted at low resolution). These Tokens, following the same procedure, re-

ceive low initial activation values and undergo chain association activation. During propagation, some of these input Tokens activate numerous similar Tokens in the memory bank due to similarity. Since these Tokens maintain close relationships with many reward and punishment symbols, the activation value chain propagation process activates numerous reward and punishment symbols (typically the owner's voiceprint features, such as unique timbre).

At this point, many reward and punishment symbols obtain activation values exceeding the preset thresholds. Assuming N reward symbols and M punishment symbols surpass these values, the machine autonomously establishes N + M targets simultaneously. In our scheme, goals are machine-generated and multi-target, rather than artificially preset through a unified reward function. All machine responses fundamentally adhere to the principle of seeking advantages while avoiding disadvantages.

After establishing these N + M targets, the machine plans its response path to increase the probability of reward symbol activation while decreasing that of punishment symbols. Thus, the machine's decisions center on achieving rewards and avoiding punishments.

The machine first processes the reward and punishment Tokens with the highest activation values, which may include one or more punishment Tokens. In the memory bank, the propagation path transmitting activation values to these punishment Tokens likely originates from the underlying feature input of the owner's voiceprint. High activation values can arise from several conditions: (1) the punishment Token possesses a very high memory value, either because it was stored with high activation (where memory value increment correlates positively with activation value) or because frequent activation through repetition elevates memory value; (2) multiple input Tokens transmit activation to this punishment Token through different pathways—for instance, the owner's "tone Tokens," "word Tokens," "master state Tokens," "expression Tokens," and "environment-related Tokens" may all connect to similar punishment symbols in memory, completing the activation value chain propagation and enabling related Tokens to obtain high activation values; (3) a tight activation value transfer relationship exists between this punishment Token and specific input Tokens, such that they consistently co-occur in memory, forming a "proximity relationship" and "high memory value relationship" with short propagation paths and high transfer coefficients. Consequently, the attention mechanism may employ both comprehensive reasoning (multiple Tokens to specific reward/punishment symbols based on integrated experience) and special-case reasoning (specific activation value transfer paths based on particular experiences). High activation punishment Tokens may also derive from activation distributions established by previous Token inputs. Although high activation values decay over time, sufficiently high values can influence machine decisions for extended periods—much like human cognition.

In this example, the propagation network comprising activation value pathways contains an immense number of Tokens that defy simple formulation. However,

language symbols typically exhibit the highest activation values (due to their frequency and high memory values). When combined according to their spatiotemporal order, the central concept may emerge as "don't lie down (cause), get scolded by owner, feel very sad (consequence)."

The machine immediately searches for the optimal response path to reduce the probability of this punishment symbol's occurrence. The decision-making principle involves increasing the probability of reward symbol activation while decreasing that of punishment symbols. Each concept forms a locally dense network in the memory bank, and the machine must reduce the probability of high-activation Tokens within this network, thereby decreasing the occurrence probability of the reward-penalty logical link.

For example, when the owner previously used similar Tokens to reprimand the machine, the system stored corresponding internal and external sensor data. Some Tokens were forgotten due to lack of repetition and memory reinforcement. However, Token combinations that repeatedly co-occurred with this "punishment symbol"—such as "lying" related Tokens, "specific time Tokens," and "specific occasion Tokens"—obtained higher memory values through repetition. Because they form recurring combinations, they mutually elevate activation values, achieving far higher memory values than non-repetitive Tokens. This positive cycle makes them easier to activate and remember, constituting the process of experience summarization.

If, in a similar environment, the owner once praised the machine, that memory would influence the decision. Thus, in comparable contexts, various Tokens may transmit activation to either punishment or reward symbols. The machine's decision represents a comprehensive statistical assessment of all potential rewards and penalties, considering both reward acquisition and punishment avoidance. When selecting a response path, the machine must differentiate between local paths leading to rewards versus those leading to punishments. This segmentation process involves adding more Tokens to the path, creating multiple segmented pathways (e.g., different scenarios, time points, or factors), enabling the machine to determine its response through these refined paths—the core of segmented imitation.

Consequently, our machine does not require conventional "Fine-tuning." Instead, it achieves "Fine-tuning" through memory accumulation, capable of performing "Fine-tuning" at any depth, in any domain, and superimposing "Fine-tuning" across countless fields without "catastrophic forgetting." This is because the system does not modify past knowledge parameters but simply amplifies the network.

In this scenario, suppose the machine is lying down during the day (conserving power, receiving rewards). After activating the punishment symbol associated with the owner's voiceprint, the machine must both avoid the activated penalty symbol's probability and increase the activated reward symbol's probability. At least two cases emerge: (1) reduce the probability of the "lying" concept

to avoid punishment (e.g., being reprimanded); (2) increase the probability of the "lying" concept to obtain rewards (e.g., power savings). The machine must make the optimal choice based on the principle of seeking advantages while avoiding disadvantages, synthesizing various response paths and comparing their statistical rewards and penalties.

If the machine is fully powered, the reward for conserving power is minimal. After completing the chain-type associative activation process, only one punishment symbol achieves high activation value. The machine will choose to avoid punishment, as this yields the highest reward value. Driven by profit maximization, the machine establishes punishment avoidance as its goal and begins constructing a response.

If the machine is low on power, power conservation becomes significant (assuming the machine must lie down to charge). After completing the chain-type associative activation process, both a punishment symbol and a reward symbol achieve high activation values. According to the principle of seeking benefits while avoiding harm, the machine simultaneously establishes two goals: achieving rewards and avoiding punishment.

The machine then creates a second-level goal: reducing the activation value of the "lying down" concept. Under this constraint, it identifies the propagation path that transfers activation to the "lying" concept and establishes a third-level objective: reducing the activation values of concepts along this path. The machine discovers that the primary path for propagating activation to the "lying" concept originates from a set of self-state sensor inputs, thus creating a third-level target: reducing the probability of these input Tokens.

The machine records various internal and external parameters during each training session, employing memory and forgetting mechanisms to encourage imitation of reward-related parameters and avoidance of punishment-related parameters. This establishes empirical connections between parameter combinations, rewards, and the internal/external environment—essentially a reinforcement learning process. Humans can also implant innate knowledge (drive-related) or directly modify machine knowledge using accumulated human experience to accelerate convergence. In different environments, environmental Tokens automatically activate the most relevant memories. By imitating these experiences, the machine passes similar parameter combinations to its motor system (including parameter types and their temporal sequences, all completed automatically), enabling it to stand up in various environments while reducing the probability of "lying" related Tokens.

When power is low, the machine's experience in achieving rewards allows it to lie down, increasing the probability of charging-related Tokens. To avoid punishment, it imitates past experience and explains its actions to the owner. The machine establishes a second-level objective: increasing the activation value achieved by the "charge" concept. To avoid punishment, it creates a third-level goal: "explaining the reasons for its behavior," because the Token combina-

tion in memory maintains a close activation value transfer relationship with the "avoid punishment" Token combination. The machine's goal thus becomes increasing the occurrence probability of this specific Token combination (explaining its behavior).

The next decision level involves activating the experience of language organization. This process iterates continuously, with new reward and penalty symbols activating each time. The machine tallies these activation values until they converge, then establishes the optimal response path.

The machine subsequently enters the imitation execution phase, decomposing its decision path iteratively down to the underlying drive parameters, issuing drive commands by imitating parameter configurations from experience. Since experience and reality can only partially match, generalization between them is achieved by imitating their common Token combination patterns.

Pathways composed of high-activation Tokens constitute top-level imitation pathways. If an imitation path lacks direct underlying driver command combinations, additional Tokens (with lower activation values) are incorporated, transforming the imitation path into multisegment pathways formed by more Tokens. When facing a large path without appropriate experience, we refine and decompose it into several smaller response path segments, seeking suitable experience for each segment. If direct underlying driver command combinations remain unattainable, we repeat the process, adding more Tokens and further decomposing the response path until it breaks down into direct underlying driver command combinations.

This process continues iteratively, with new Tokens potentially arriving at any moment. Whenever new Tokens are input, the machine must repeat the chain association activation process. Upon completion, the activation value distribution in the memory bank changes, prompting the machine to restart its decision-making process. Consequently, the machine's optimal decision may involve abandoning some current goals to pursue newly emerged objectives.

Thus, our machine generates its own goals and can continuously modify them, resulting in highly flexible, environment-adaptive decision-making. In the above example, the machine might immediately stand up, enhance its sound processing system resolution, and turn to observe the owner's posture, movements, and expressions—even before the owner has finished saying "Hello…"

Our machine embodies human-like intelligence whose information understanding derives from personal experience rather than statistical processes. Only through this approach can machines deliver truly personalized services. A thousand housewives have a thousand distinct requirements; AI systems based on knowledge statistics, with robots incapable of real-time knowledge updates, can never truly enter homes or win hearts. Their application scenarios remain severely limited. Our solution represents genuine general artificial intelligence and may fundamentally transform the world.

9 Conclusion

We contend that artificial intelligence development can be roughly divided into stages: (1) The "feature exploration" stage—primarily manual exploration before deep learning, shifting to machine exploration afterward. (2) Following the realization of true attention mechanisms (Transformer), machines achieve "knowledge generalization" after initial alignment between the machine's "knowledge coordinate basis cluster" and the human "knowledge coordinate basis cluster (concepts)." When confronting human tasks, machines demonstrate certain intelligence through "knowledge generalization" [29][30]. One-dimensional attention mechanisms enable large language models, two-dimensional attention enables image generalization, and three-dimensional attention can achieve 3D creative capabilities. Four-dimensional (3D + time) attention mechanisms can generalize dynamic processes, enabling video generation and robot services in limited scenarios.

However, we believe that only by incorporating "vitality—the fifth dimension, self-demand" can we truly instill a "soul" into machine intelligence. While large models are destined to achieve this "fifth dimension," our solution can endow machines with "life," enabling them to become genuine "universal artificial intelligence."

Therefore, we propose that AI must advance to the next stage: the "autonomous interaction" stage. "Autonomy" signifies that machines are no longer silent "machines" but can spontaneously generate behaviors (equivalent to self-programming) and actively explore knowledge (e.g., by interacting with the environment). "Interaction" means machines can engage with the environment in real time, update knowledge continuously, and make sequential decisions to accomplish complex tasks in unfamiliar environments [29].

Many distinguished scholars have proposed perspectives on achieving true general artificial intelligence. Professor LeCun introduced the "world model," while Professor Zhu Songchun outlined four characteristics of general AI: (1) ability to perform unlimited tasks, (2) ability to independently generate new tasks, (3) value system-driven operation, and (4) possession of a world model reflecting reality. Our plan directly responds to the visions of both Professor LeCun and Professor Zhu.

General artificial intelligence represents both the original aspiration and the ultimate crown of AI research. We present a comprehensive technical solution for implementing general AI, including detailed step-by-step implementation procedures. In references [25][26][27][28], we disclose the technical steps to achieve this path through patents, which may represent the correct path toward leading humanity to general artificial intelligence.

References

[1] Reed S, Zolna K, Parisotto E, et al. A Generalist Agent[J]. 2022. [2]

Optimizing Language Models for Dialogue. https://openai.casa/blog/chatgpt/ [3] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[J]. arXiv e-prints, 2022. [4] Vinyals O, Ewalds T, Bartunov S, et al. StarCraft II: A New Challenge for Reinforcement Learning[J]. 2017. [5] Quanjun Zhang, Chunrong Fang, Yuxiang Ma, Weisong Sun, Zhenyu Chen. A Survey of Learning-based Automated Program Repair. arXiv:2301.03270, 2023 [6] Antonio Mastropaolo, Luca Pascarella, Emanuela Guglielmi, Matteo Ciniselli, Simone Scalabrino, Rocco Oliveto, Gabriele Bavota. On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot. arXiv:2302.00438, 2023 [7] Introducing ChatGPT. https://openai.com/blog/chatgpt/ [8] Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, Lijuan Wang. arXiv:2210.09150, 2023 [9] An experimental open-source attempt to make GPT-4 fully autonomous. https://GitHub.com/Significant-Gravitas/Auto-GPT [10] Zhuosheng Zhang, Aston Zhang, Mu Li, Alex Smola. Automatic Chain of Thought Prompting in Large Language Models. arXiv:2210.03493, 2023 [11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022), 2022a. https://arxiv.org/abs/2201.11903 [12] AUTO GPT, https://auto-gpt.ai/ [13] ERNIE Bot, https://mp.weixin.qq.com/s/0-8X9FPouteKzNiK6DPaiA [14] Mccann B, Bradbury J, Xiong C, et al. Learned in Translation: Contextualized Word Vectors[J]. 2017. [15] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. arXiv, 2017. [16] Cheng J, Dong L, Lapata M. Long Short-Term Memory-Networks for Machine Reading[J]. 2016. [17] Lin Z, Feng M, Santos C, et al. A Structured Self-attentive Sentence Embedding[J]. 2017. [18] Parikh A, Tckstrm O, Das D, et al. A Decomposable Attention Model for Natural Language Inference[J]. 2016. [19] Paulus R, Xiong C, Socher R. A Deep Reinforced Model for Abstractive Summarization[J]. 2017. [20] Tamkin A, Brundage M, Clark J, et al. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models[J]. 2021. [21] Kodge S, Roy K. BERMo: What can BERT learn from ELMo?[J]. 2021. [22] Lee H, Xu G. Optimizing Policy via Deep Reinforcement Learning for Dialogue Management[C]// 2018 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE Computer Society, 2018. [23] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[J]. arXiv e-prints, 2022. [24] Sunehag P, Hutter M. Principles of Solomonoff Induction and AIXI[C]// Ray Solomonoff memorial conference. Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia; Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia, Department of Computer Science, ETH Zurich, Switzerland, 2011. [25] Yongcong Chen, Ting Zeng, Xingyue Chen. Method for implementing universal artificial intelligence. CN111553467A[P]. 2020. [26] Yongcong Chen, Ting Zeng, Xingyue Chen. A Machine Intelligence Implementation Method Imitating Human Intelligence, CN111582457A[P]. 2020. [27] Yongcong Chen,

Ting Zeng, Xingyue Chen. Establishment of artificial general intelligence system, CN112016664A[P]. 2020. [28] Chen Y, Zeng T, Chen X. ESTABLISHMENT OF GENERAL-PURPOSE ARTIFICIAL INTELLIGENCE SYSTEM:, US2022121961A1[P]. 2022. [29] Yongcong Chen, Ting Zeng, Xiaoyi Qian. A Preliminary Study on the Capability Boundary of LLM and a New Implementation Approach for AGI. ChinaXiv:202305.00046 [30] Yongcong Chen, Ting Zeng, Jun zhang. A New Solution for Realizing True Artificial General Intelligence. ChinaXiv:202308.00026

Note: Figure translations are in progress. See original paper for figures.